

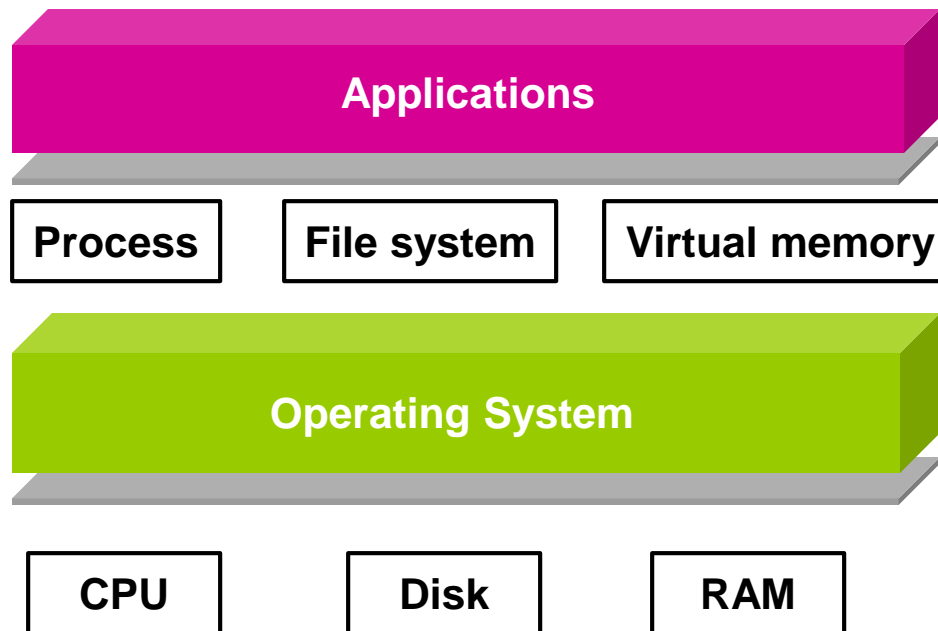
**CSE 153**  
**Design of Operating  
Systems**

**Fall 2018**

Lecture 12: File Systems (1)—Disk drives

# OS Abstractions

---



# File Systems Agenda

---

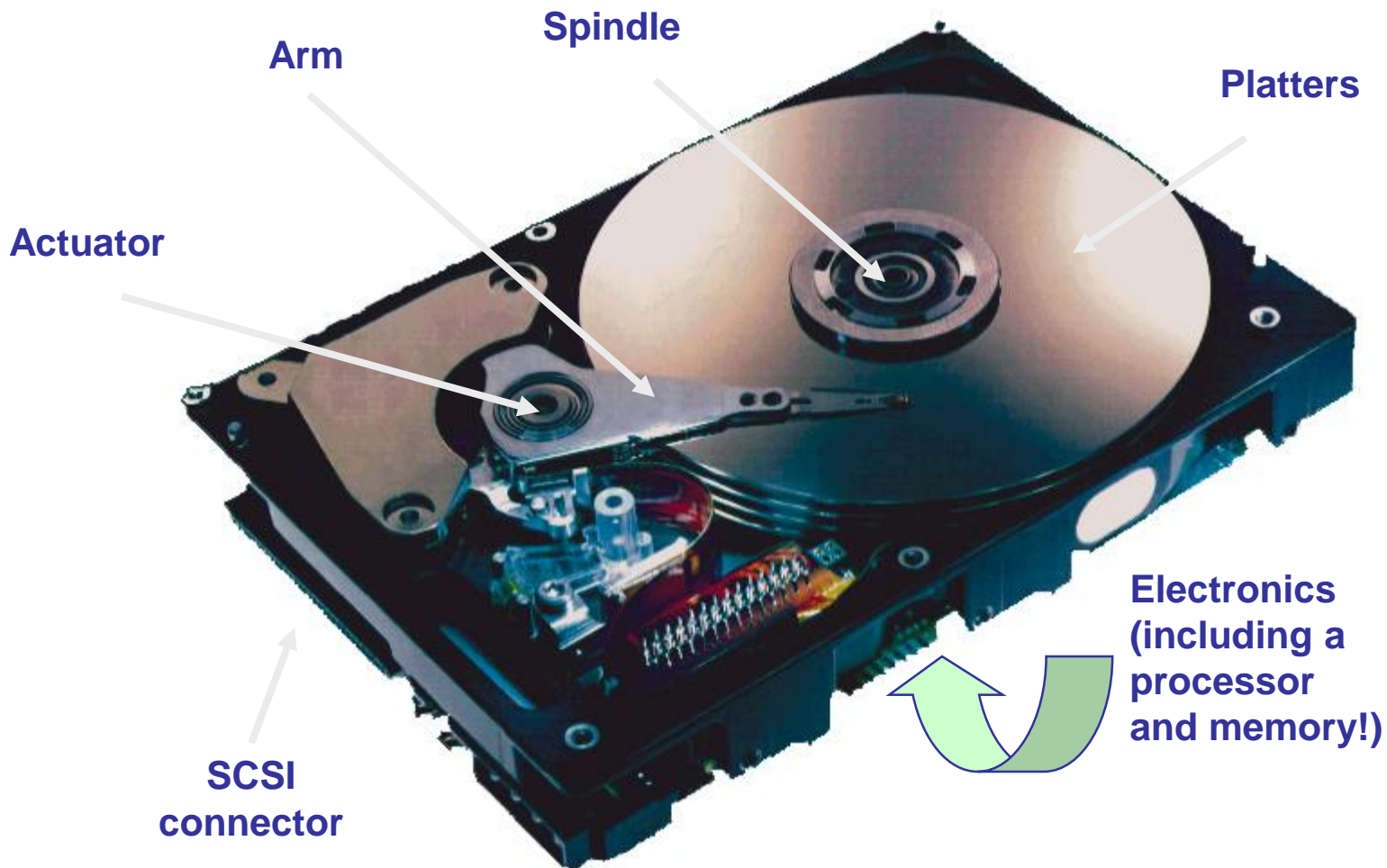
- First we'll discuss properties of physical disks
  - ◆ Structure, Performance
  - ◆ Scheduling
- Then we'll discuss how to build file systems (next time)
  - ◆ Abstraction:
    - » Files
    - » Directories
    - » Sharing
    - » Protection
  - ◆ Implementation
    - » File System Layouts
    - » File Buffer Cache
    - » Read Ahead

# Disks and the OS

---

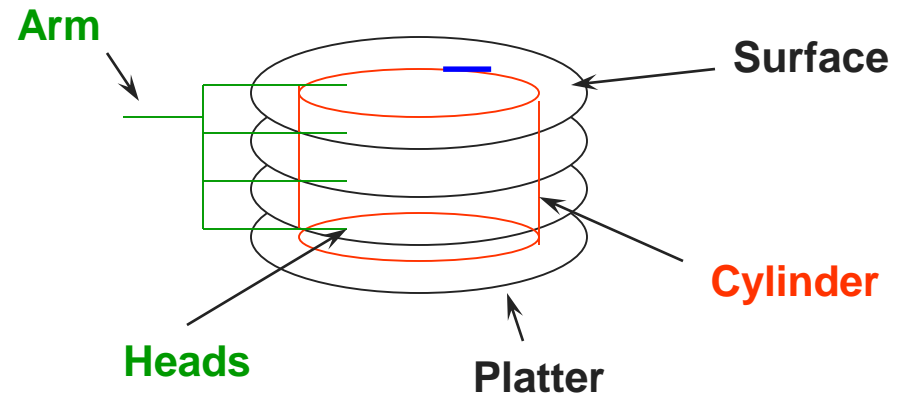
- Disks are messy physical devices:
  - ◆ Errors, bad blocks, missed seeks, etc.
- OS' s job is to hide this mess from higher level software
  - ◆ Low-level device control (initiate a disk read, etc.)
- Os provides Higher-level abstractions (files, databases, etc.)
  - ◆ OS maps them to the device and implements policies to promote
    - » Performance, reliability, protection, ...

# What's Inside A Disk Drive?

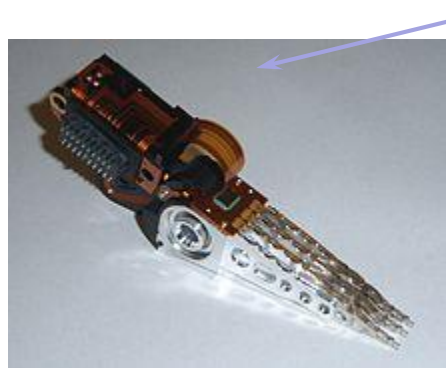
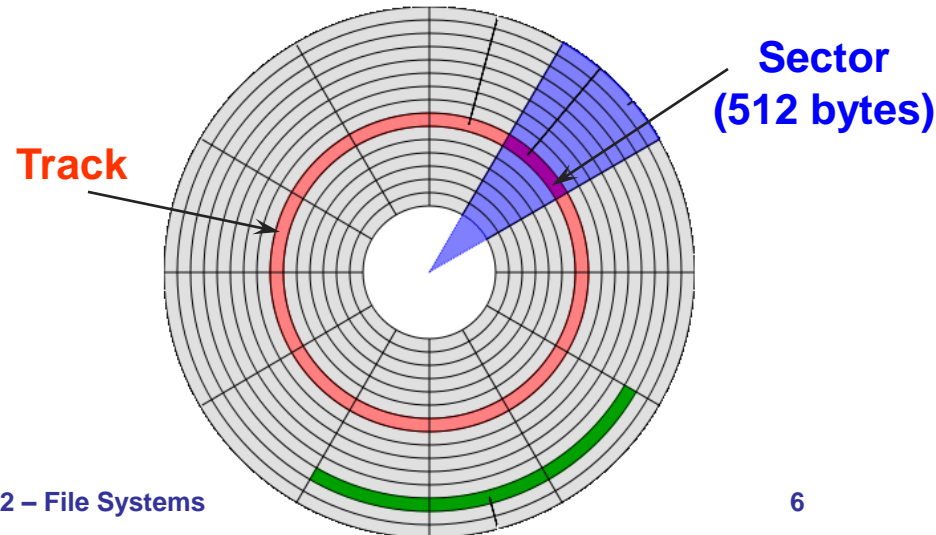


# Physical Disk Structure

- Disk components
  - ◆ Platters
  - ◆ Surfaces
  - ◆ Tracks
  - ◆ Sectors
  - ◆ Cylinders
  - ◆ Arm
  - ◆ Heads

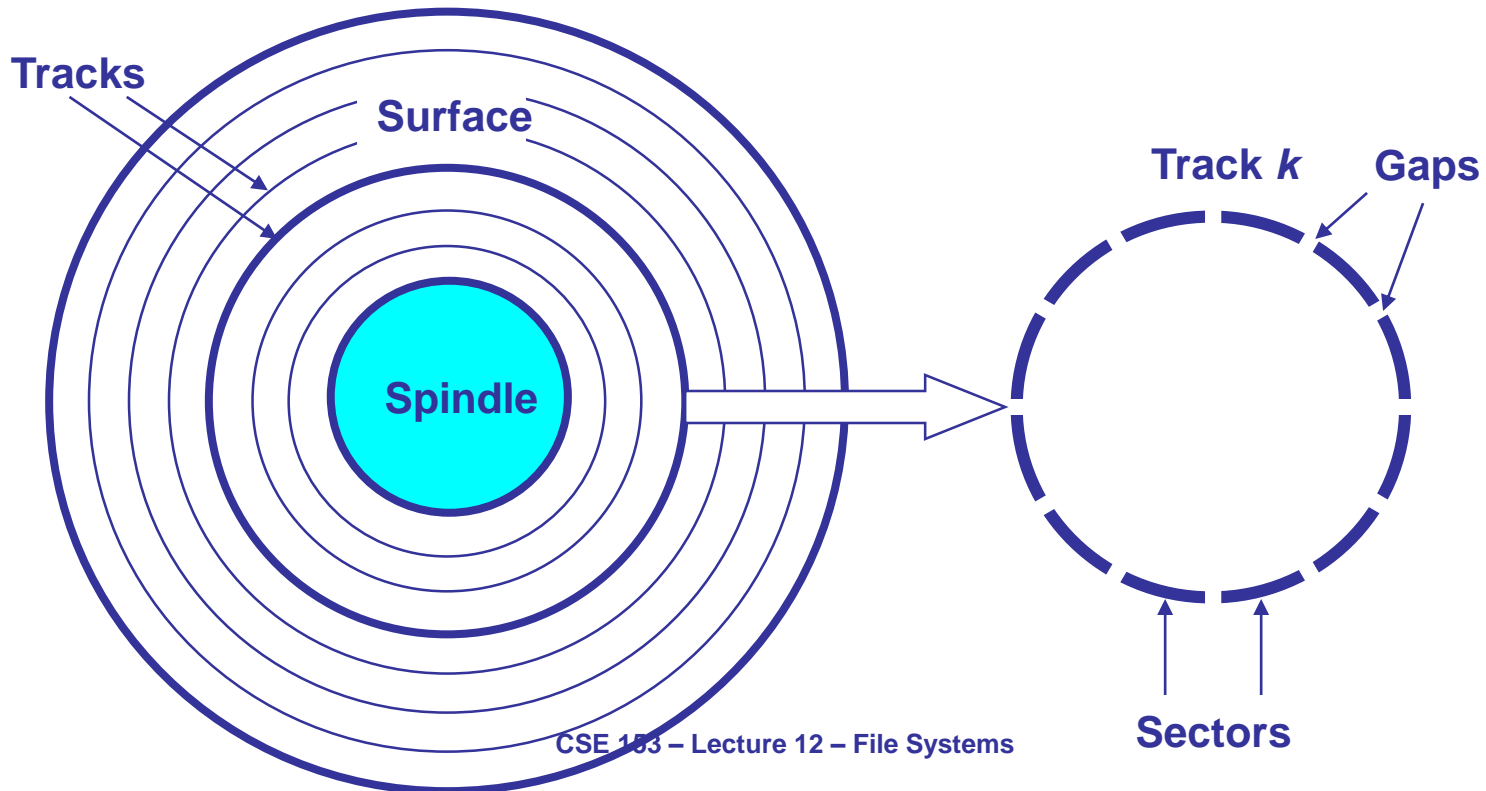


**Arm**



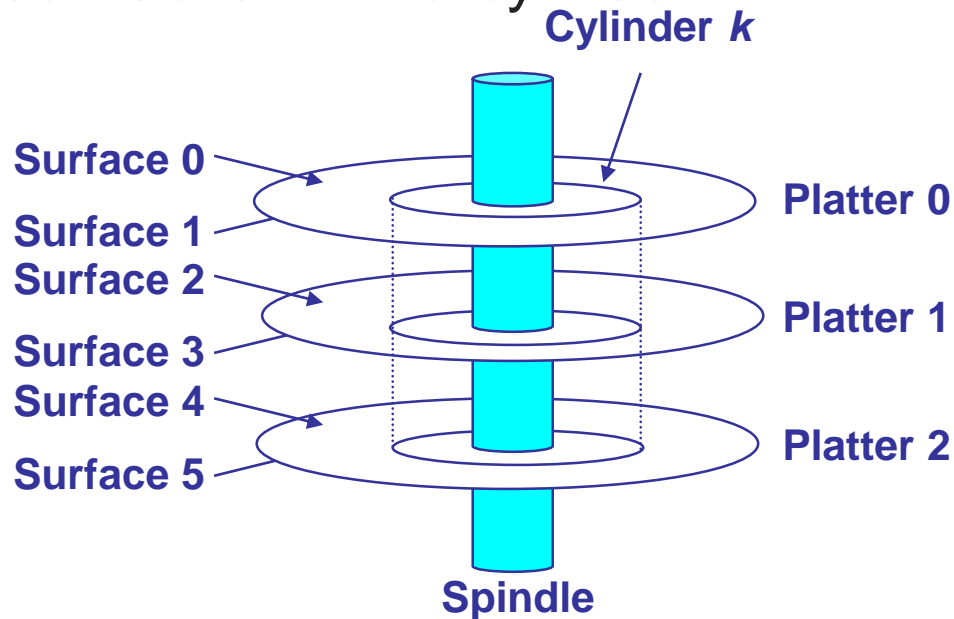
# Disk Geometry

- Disks consist of **platters**, each with two **surfaces**.
- Each surface consists of concentric rings called **tracks**.
- Each track consists of **sectors** separated by **gaps**.



# Disk Geometry (Multiple-Platter View)

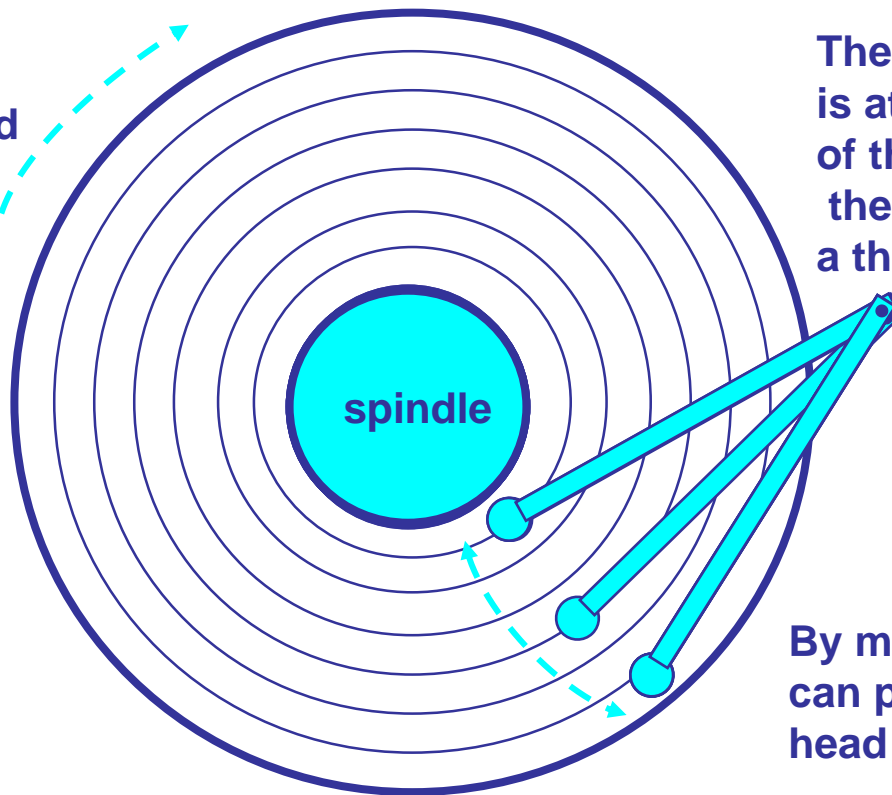
- Aligned tracks form a cylinder.





# Disk Operation (Single-Platter View)

The disk surface spins at a fixed rotational rate

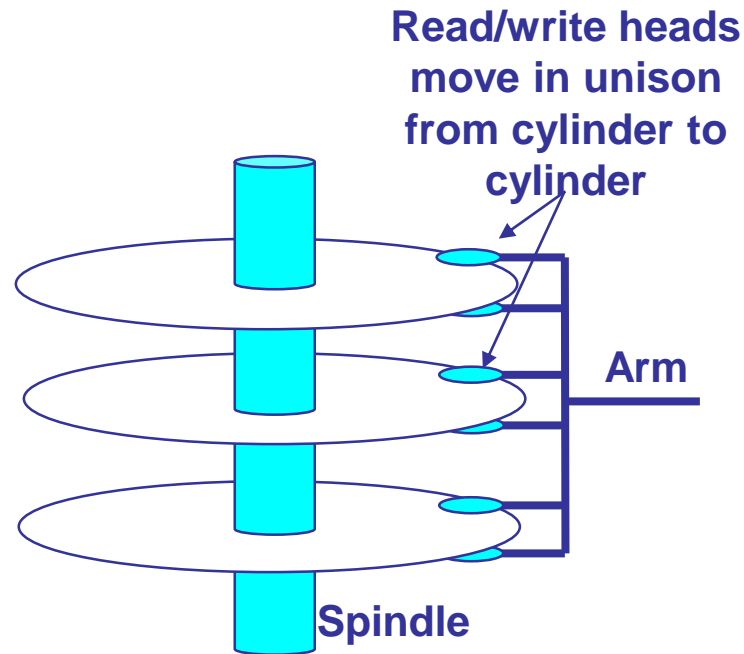


The read/write *head* is attached to the end of the *arm* and flies over the disk surface on a thin cushion of air.

By moving radially, the arm can position the read/write head over any track.

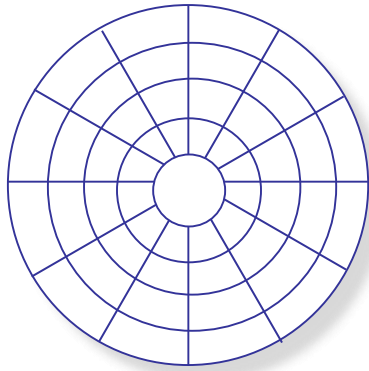
# Disk Operation (Multi-Platter View)

---



# Disk Structure - top view of single platter

---

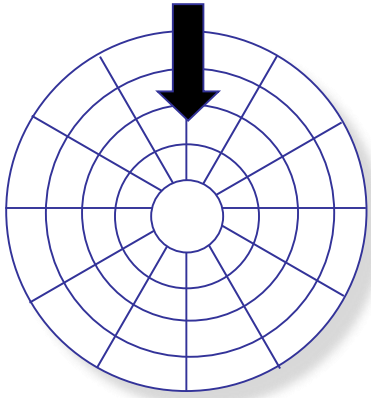


**Surface organized into tracks**

**Tracks divided into sectors**

# Disk Access

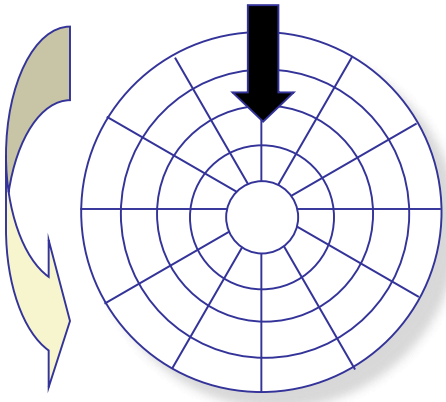
---



**Head in position above a track**

# Disk Access

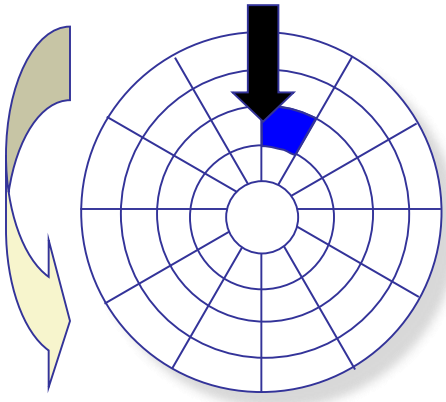
---



**Rotation is counter-clockwise**

# Disk Access – Read

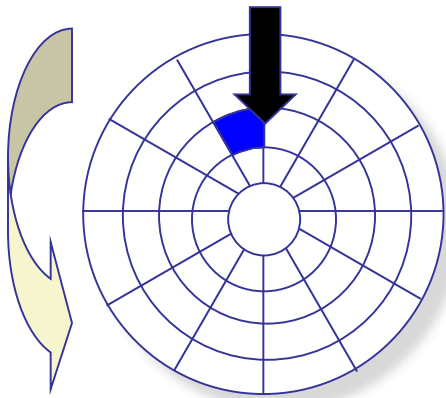
---



**About to read blue sector**

# Disk Access – Read

---

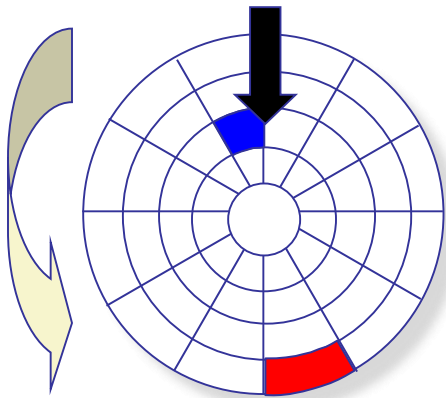


After **BLUE**  
read

**After reading blue sector**

# Disk Access – Read

---



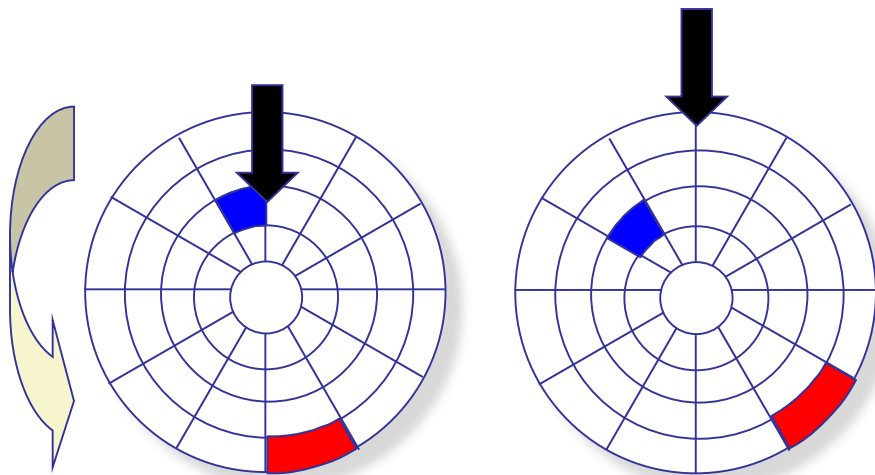
After **BLUE**  
read

**Red request scheduled next**



# Disk Access – Seek

---



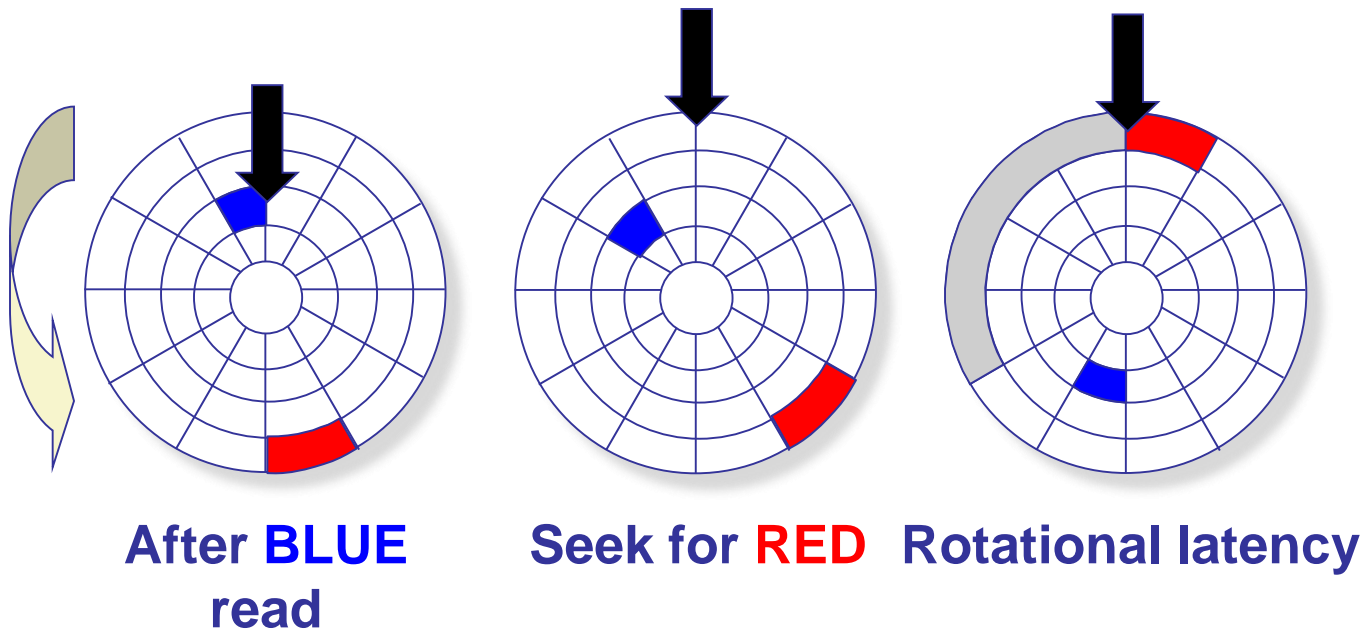
After **BLUE**  
read

Seek for **RED**

**Seek to red's track**

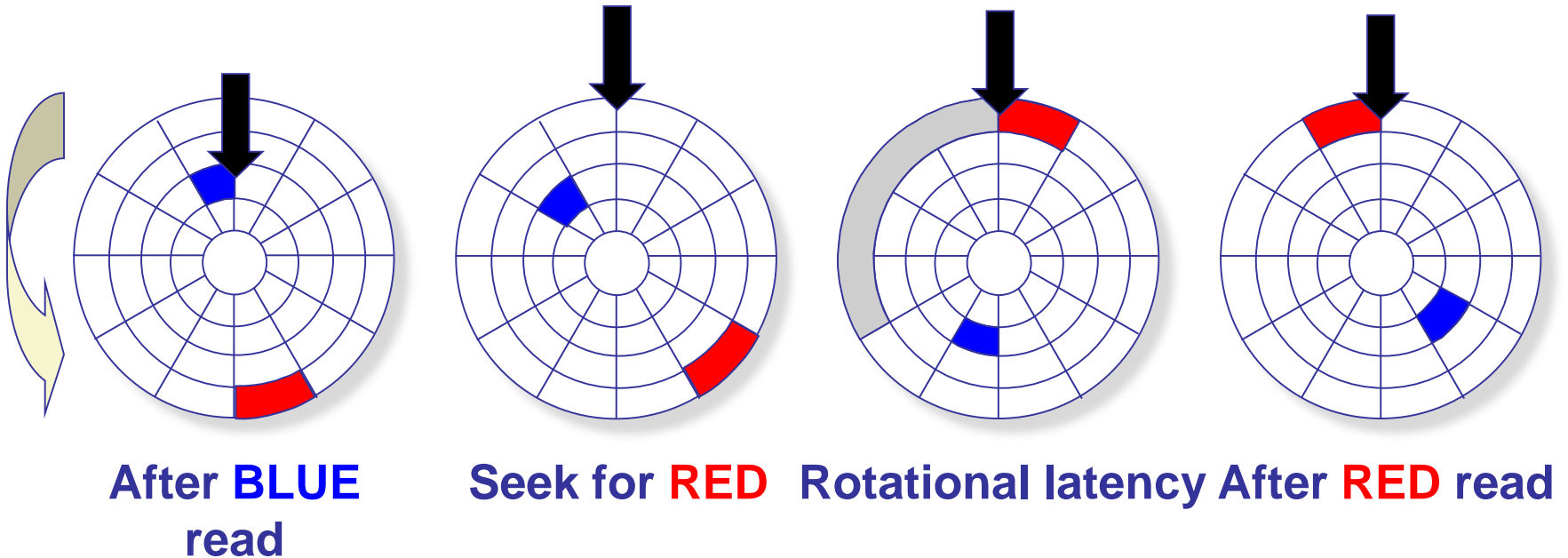
# Disk Access – Rotational Latency

---



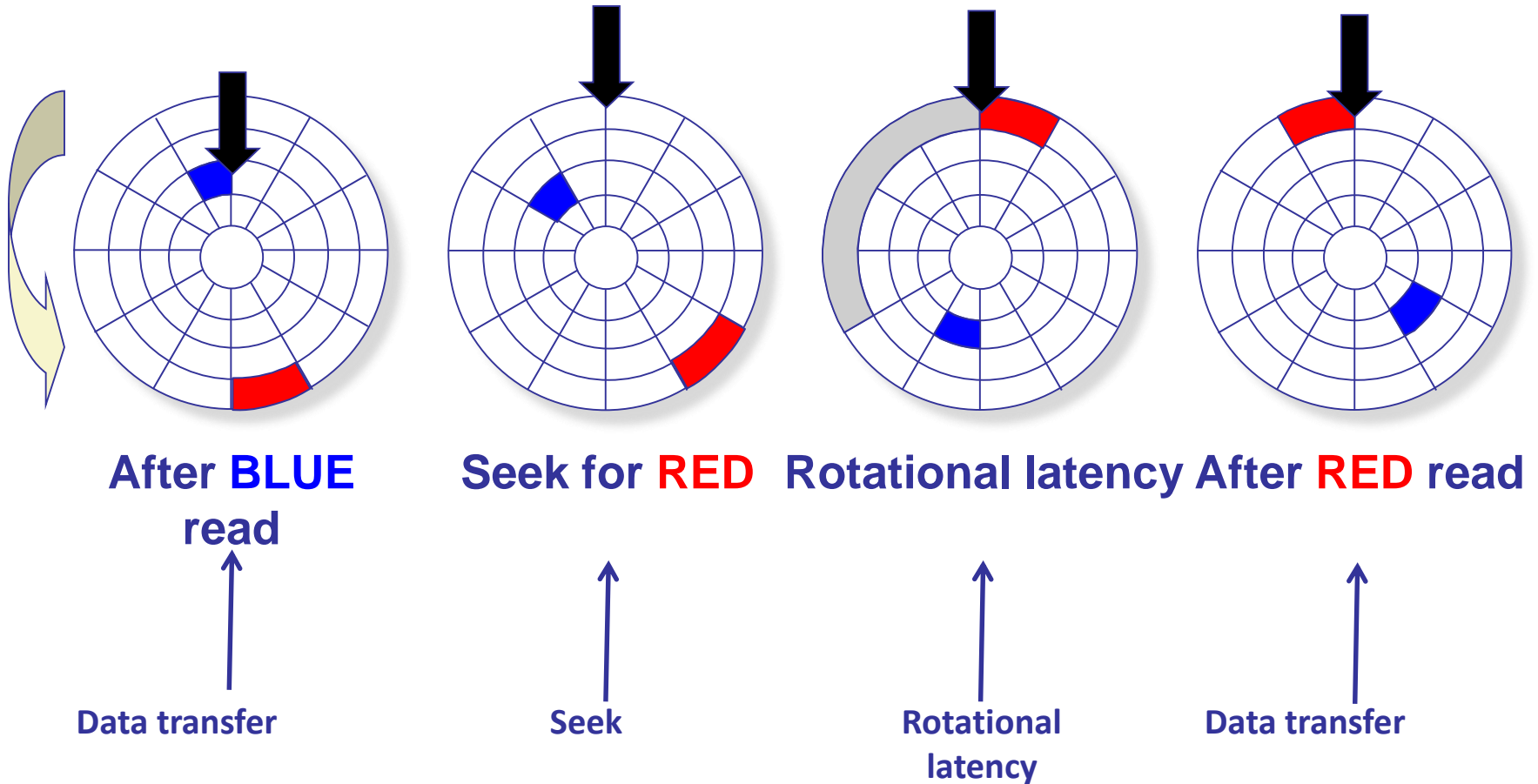
**Wait for red sector to rotate around**

# Disk Access – Read



**Complete read of red**

# Disk Access – Service Time Components



# Disk Access Time

---

- Average time to access some target sector approximated by :
  - ◆  $T_{\text{access}} = T_{\text{avg seek}} + T_{\text{avg rotation}} + T_{\text{avg transfer}}$
- **Seek time** ( $T_{\text{avg seek}}$ )
  - ◆ Time to position heads over cylinder containing target sector.
  - ◆ Typical  $T_{\text{avg seek}}$  is 3—9 ms
- **Rotational latency** ( $T_{\text{avg rotation}}$ )
  - ◆ Time waiting for first bit of target sector to pass under r/w head.
  - ◆  $T_{\text{avg rotation}} = 1/2 \times 1/\text{RPMs} \times 60 \text{ sec}/1 \text{ min}$
  - ◆ Typical  $T_{\text{avg rotation}} = 7200 \text{ RPMs}$
- **Transfer time** ( $T_{\text{avg transfer}}$ )
  - ◆ Time to read the bits in the target sector.
  - ◆  $T_{\text{avg transfer}} = 1/\text{RPM} \times 1/(\text{avg \# sectors/track}) \times 60 \text{ secs}/1 \text{ min.}$

# Disk Access Time Example

---

- Given:
  - ◆ Rotational rate = 7,200 RPM
  - ◆ Average seek time = 9 ms.
  - ◆ Avg # sectors/track = 400.
- Derived:
  - ◆  $T_{\text{avg rotation}} = 1/2 \times (60 \text{ secs}/7200 \text{ RPM}) \times 1000 \text{ ms/sec} = 4 \text{ ms}.$
  - ◆  $T_{\text{avg transfer}} = 60/7200 \text{ RPM} \times 1/400 \text{ secs/track} \times 1000 \text{ ms/sec} = 0.02 \text{ ms}$
  - ◆  $T_{\text{access}} = 9 \text{ ms} + 4 \text{ ms} + 0.02 \text{ ms}$
- Important points:
  - ◆ Access time dominated by seek time and rotational latency.
  - ◆ First bit in a sector is the most expensive, the rest are free.
  - ◆ SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
    - » Disk is about 40,000 times slower than SRAM,
    - » 2,500 times slower than DRAM.

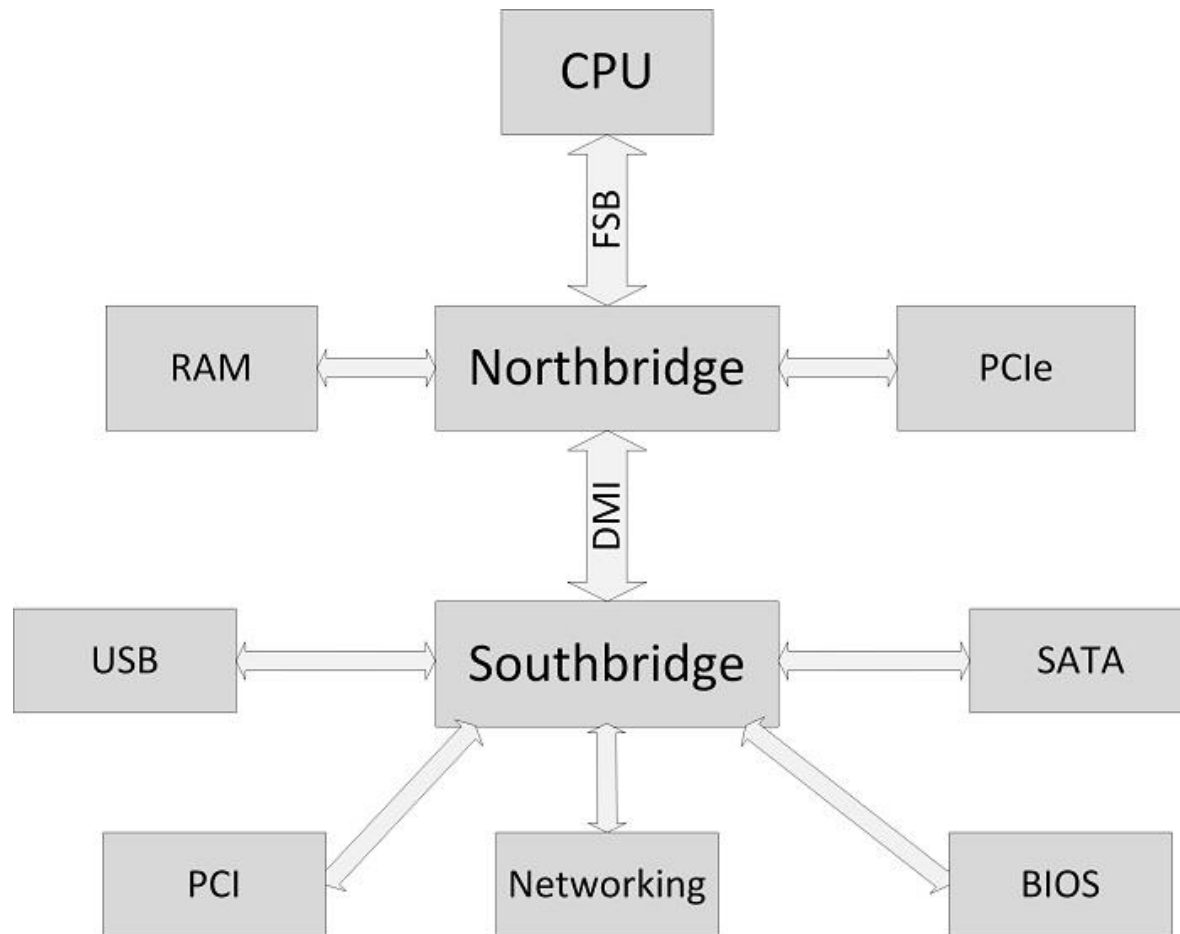
# Logical Disk Blocks

---

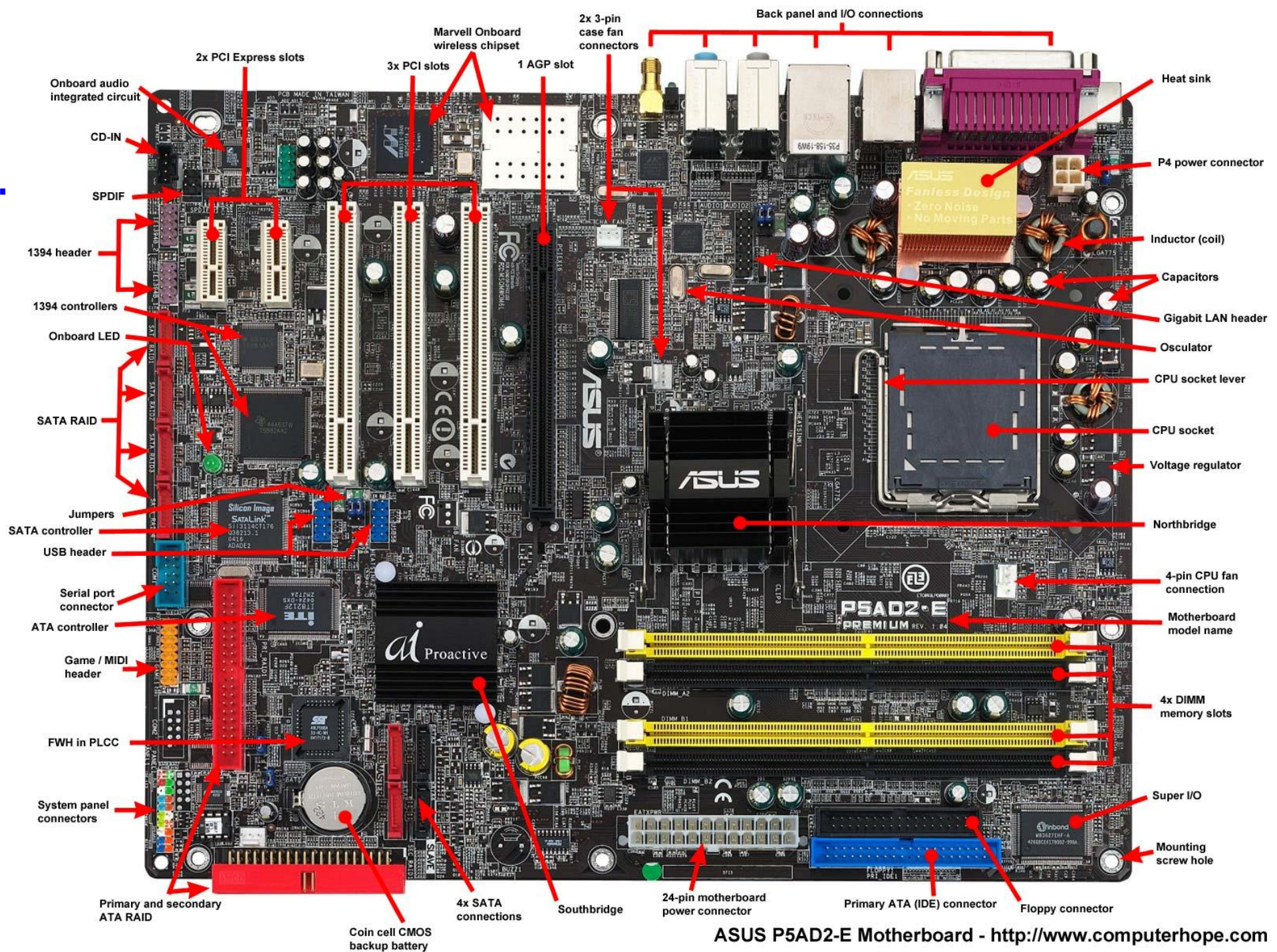
- Modern disks present a simpler abstract:
  - ◆ The set of available sectors is modeled as a sequence of  $b$ -sized **logical blocks** (0, 1, 2, ...)
- Mapping between logical and actual (physical) sectors
  - ◆ Maintained by a device called disk controller.
  - ◆ Converts requests for logical blocks into (surface, track, sector)
  - ◆ Allows controller to set aside spare cylinders for each zone.
  - ◆ Accounts for the difference in “formatted capacity” and “maximum capacity”.

# I/O and disk in the system

---

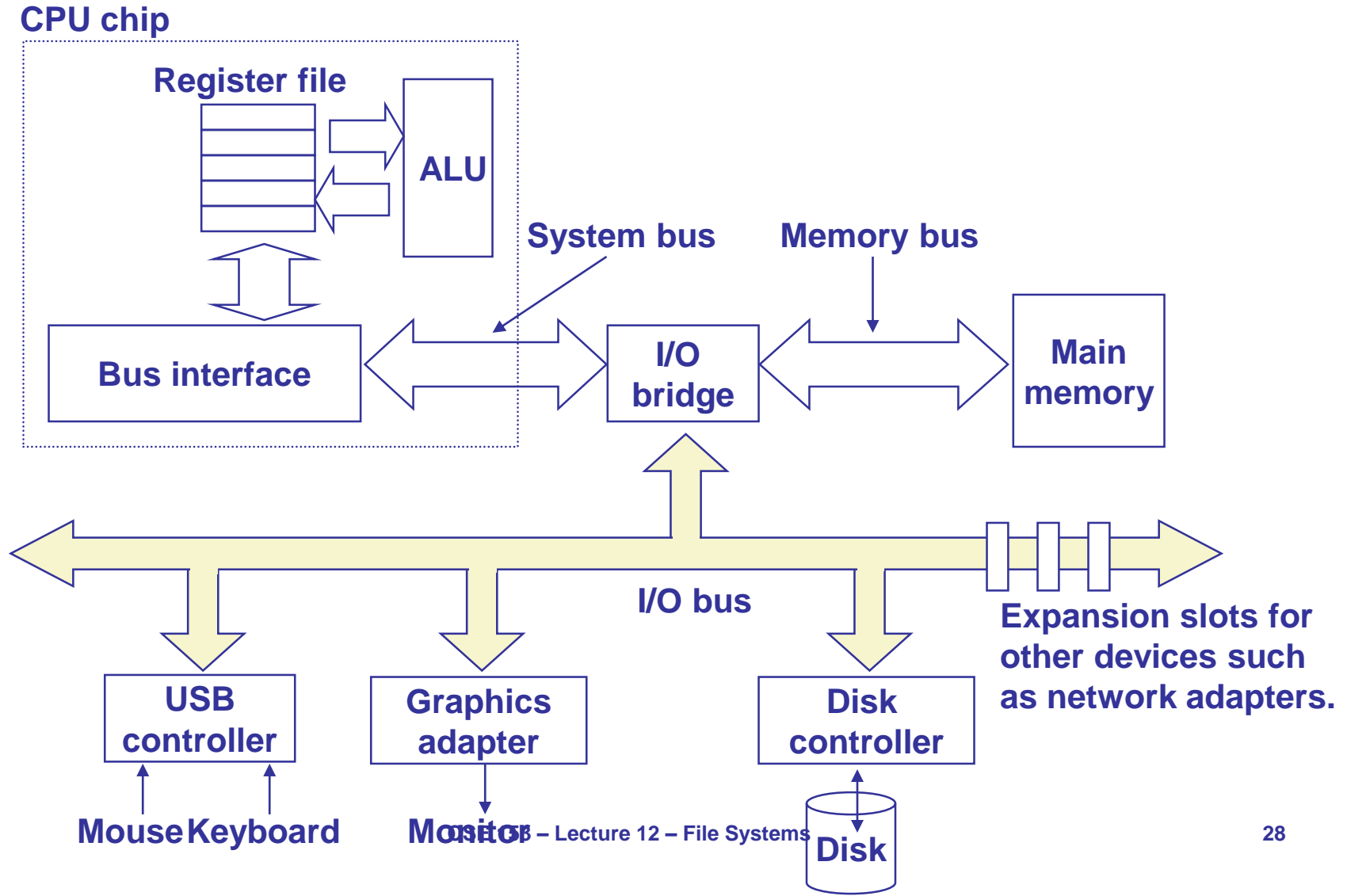






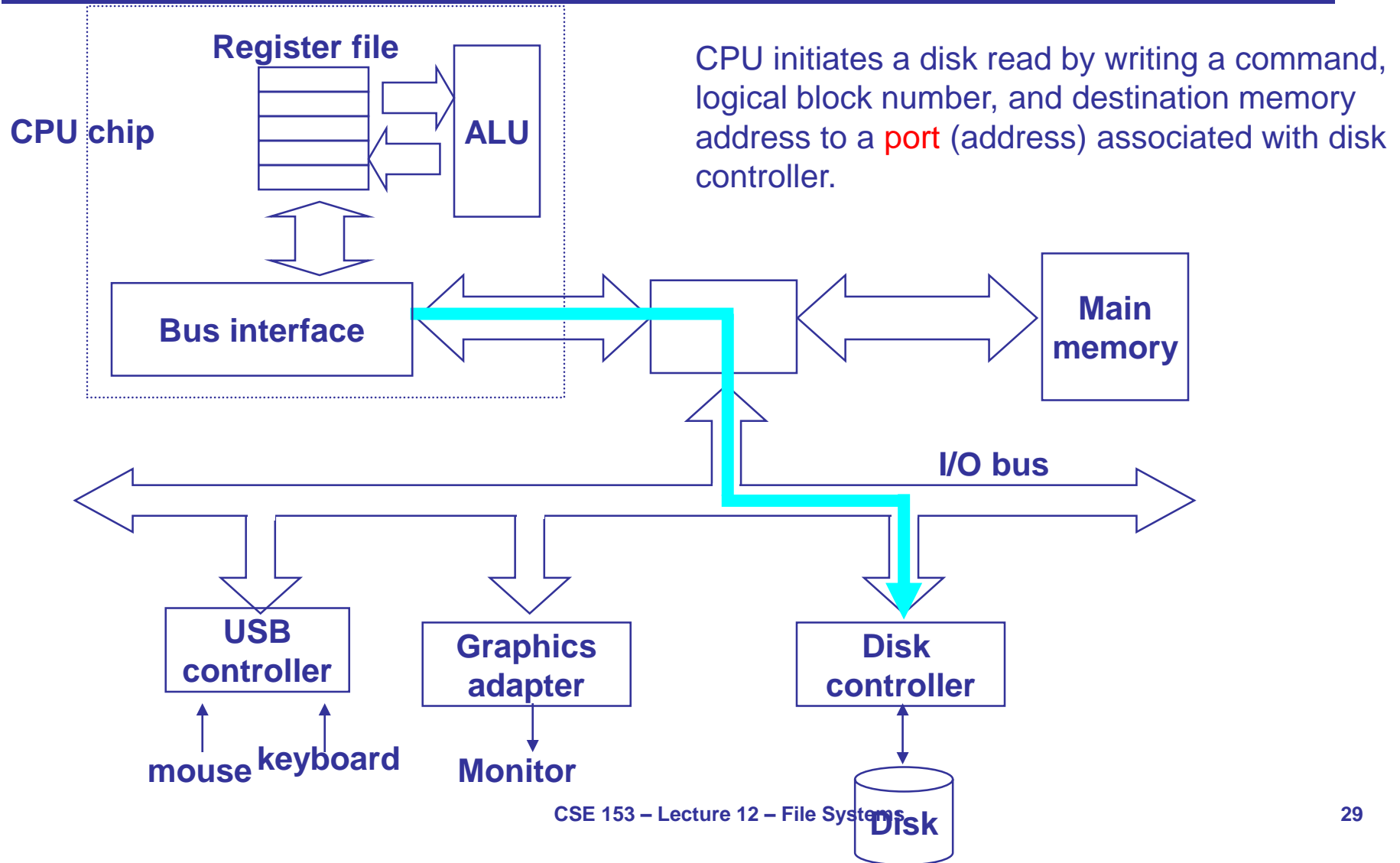
ASUS P5AD2-E Motherboard - <http://www.computerhope.com>

# I/O Bus

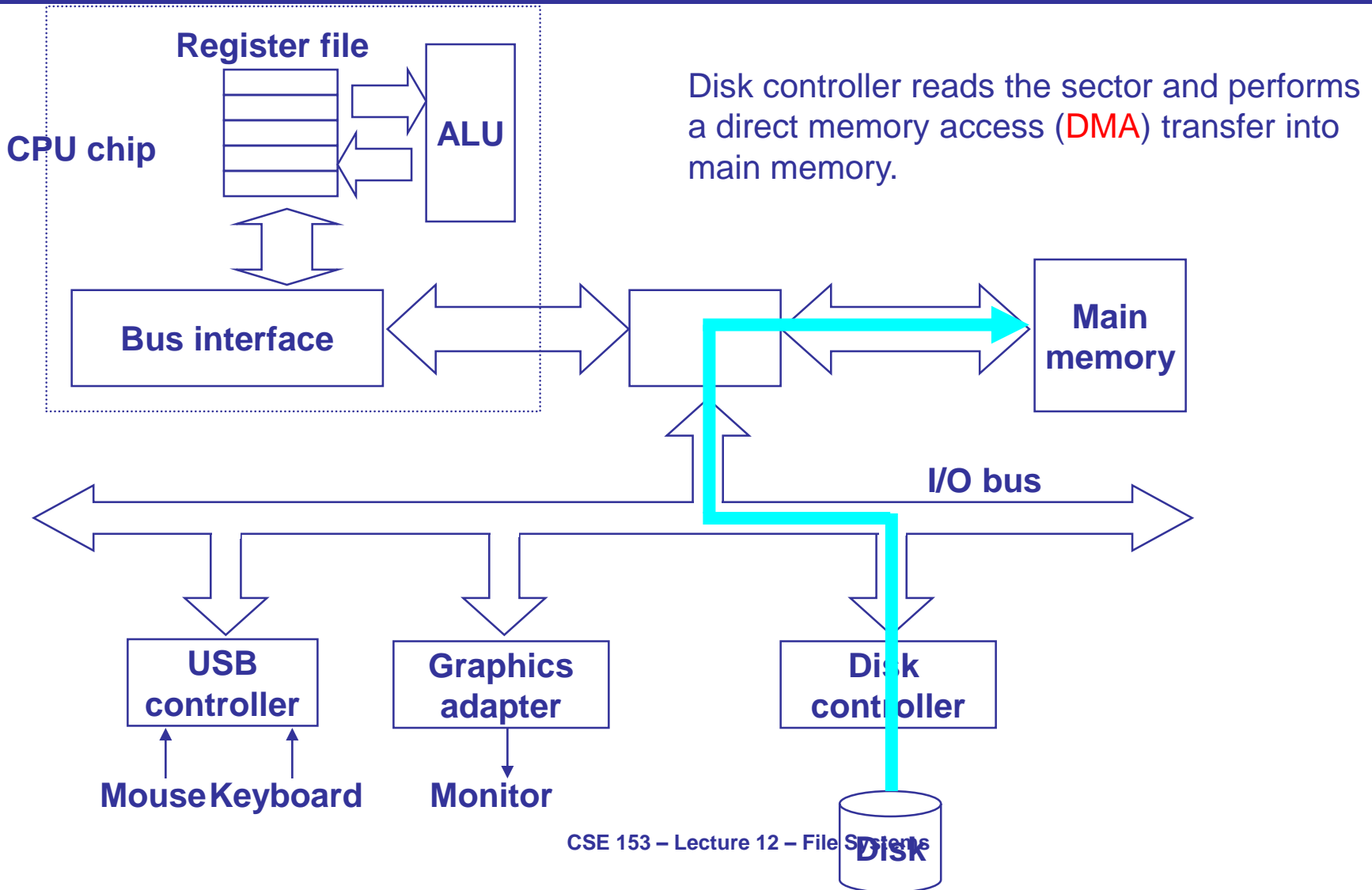




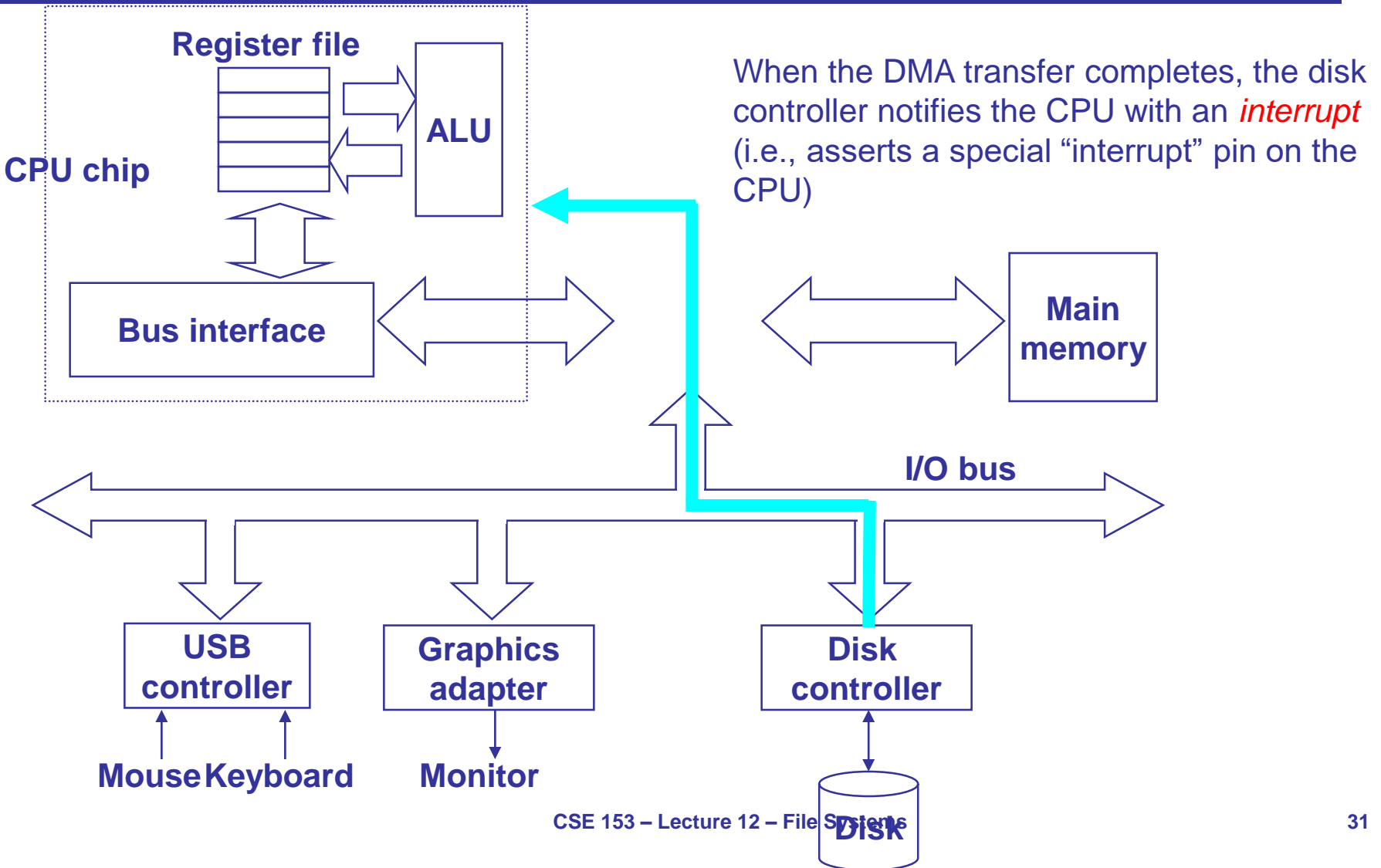
# Reading a Disk Sector (1)



# Reading a Disk Sector (2)



# Reading a Disk Sector (3)



# Disks Heterogeneity

---

- Seagate Barracuda 3.5" ([workstation](#))
  - ◆ capacity: 250 - 750 GB
  - ◆ rotational speed: 7,200 RPM
  - ◆ sequential read performance: 78 MB/s (outer) - 44 MB/s (inner)
  - ◆ seek time (average): 8.1 ms
- Seagate Cheetah 3.5" ([server](#))
  - ◆ capacity: 73 - 300 GB
  - ◆ rotational speed: 15,000 RPM
  - ◆ sequential read performance: 135 MB/s (outer) - 82 MB/s (inner)
  - ◆ seek time (average): 3.8 ms
- Seagate Savvio 2.5" ([smaller form factor](#))
  - ◆ capacity: 73 GB
  - ◆ rotational speed: 10,000 RPM
  - ◆ sequential read performance: 62 MB/s (outer) - 42 MB/s (inner)
  - ◆ seek time (average): 4.3 ms

# Recent: Seagate Enterprise (2015)



[Press Resources](#) > [Press Release](#)

[Products](#) > [Semiconductors](#)

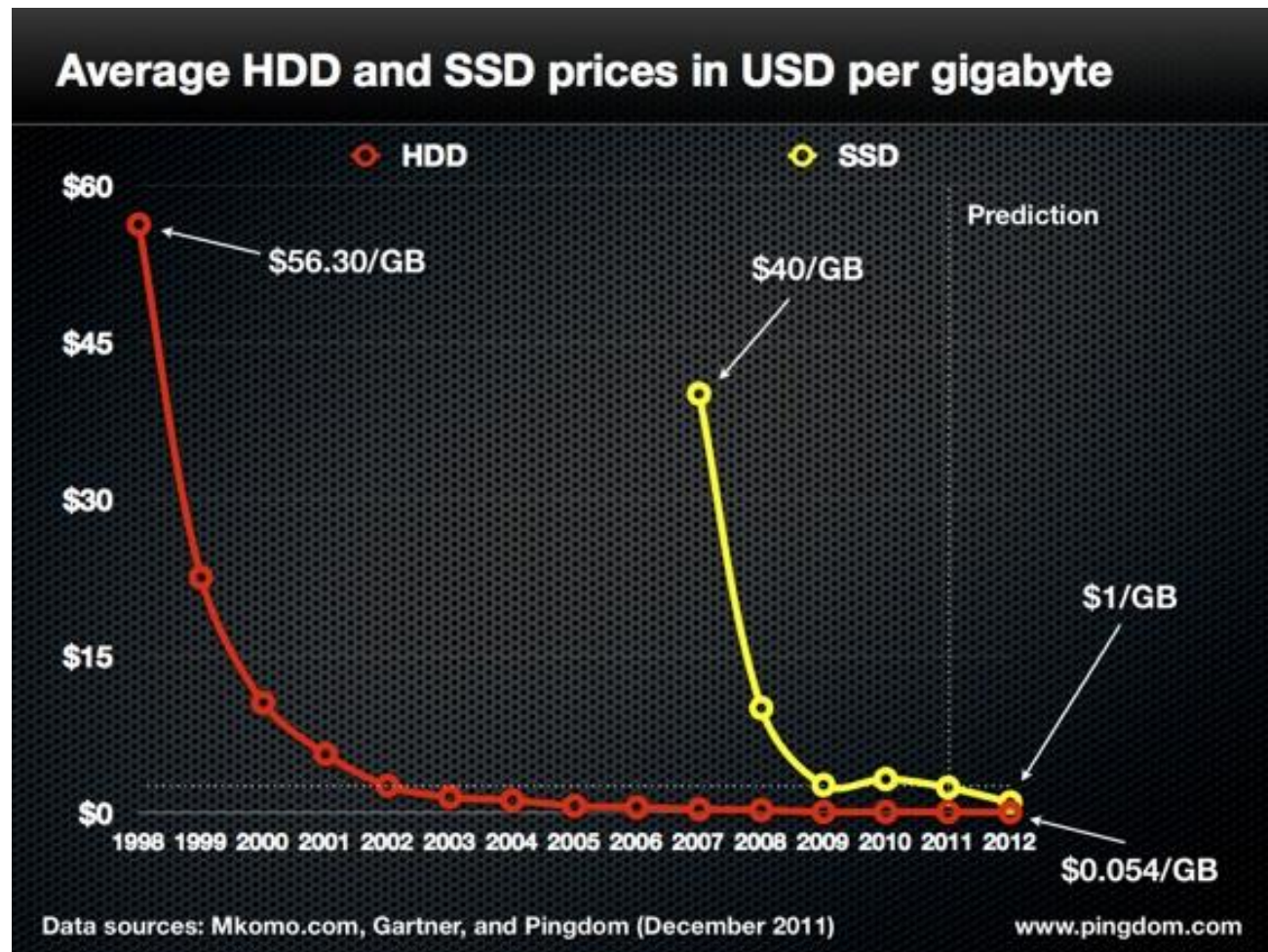
## Samsung Electronics Begins Mass Production of Industry's Largest Capacity SSD – 30.72TB – for Next-Generation Enterprise Systems

Korea on February 20, 2018



*New 'PM1643' is built on latest 512Gb V-NAND to offer the most advanced storage, featuring industry-first 1TB NAND flash package, 40GB of DRAM, new controller and custom software*

# Contrarian View



- A lot of file system ideas tied to disk drives are no longer relevant?



# Disk Scheduling

---

- Because seeks are so expensive (milliseconds!), OS schedules requests that are queued waiting for the disk
  - ◆ **FCFS** (do nothing)
    - » Reasonable when load is low
    - » **Does nothing to minimize overhead of seeks**
  - ◆ **SSTF** (shortest seek time first)
    - » Minimize arm movement (seek time), maximize request rate
    - » **Favors middle blocks, potential starvation of blocks at ends**
  - ◆ **SCAN** (elevator)
    - » Service requests in one direction until done, then reverse
    - » **Long waiting times for blocks at ends**
  - ◆ **C-SCAN**
    - » Like SCAN, but only go in one direction (typewriter)

# Disk Scheduling (2)

---

- In general, unless there are request queues, disk scheduling does not have much impact
  - ◆ Important for servers, less so for PCs
- Modern disks often do the disk scheduling themselves
  - ◆ Disks know their layout better than OS, can optimize better
  - ◆ Ignores, undoes any scheduling done by OS